



State of Montana Project Management Office

Project Execution and Approval Phase

User Acceptance Test Plan Instructions

Template for a User Acceptance Test Plan to be used for final “approval” of the system’s state by customers before deploying an application to users. Such a plan typically describes the purpose and scope of a particular set of testing; the team to perform the testing, and their responsibilities; and specific test scenarios, environment and conditions.

User Acceptance Testing is a key feature of preparing for implementation; it provides a formal means for ensuring that a new system actually meets the necessary user requirements, from the user or customer’s perspective. UAT is intended to test the system in as close to a full normal day-to-day operational environment as possible before deploying the system for general use. The testing can be broken down into tests of specific modules of the system functionality and its support documentation. It should take into account both functionality and usability.

The tests and the procedure by which testing is performed and results reviewed and addressed, must ensure that the system or application is compliant with business rules, meets the users' expectations and performs as expected in the actual business environment. The UAT should, therefore, check different aspects of the system such as:

- Sufficient degree of detail, for instance, on Help screens.
- Complete screens – all necessary information to perform a business process or user transaction is present.
- Correct content from the user's point of view – quality and accuracy of data.
- Usable results (screens returned after an operation, reports generated, etc., including layout and contents).
- System performs the business functions as required.
- Accuracy and usefulness and usability of user documentation and procedures.
- Release, installation and configuration management procedures.

Follow these steps to prepare for and execute the User Acceptance Tests:

1. Involve people from the user or customer community who understand the business processes, customer-view transactions and overall functions the system must perform, to help identify tests to be run.

2. Train the users if necessary on how to use the software, how to run tests and how to record results and defects. (See related notes, however, in the Training section of the outline starting on page 3 of this document.)
3. When the system is ready for release to acceptance testing, the users should execute functional tests, as well as help run tests of installation and online help and review any written user documentation.
4. Determine the 'defect rating' scale to be used to categorize the problems found during test, to help with prioritization of correction work and decisions as to which items must be corrected before release to general use.
5. Have all testers document the results of each test, and provide error reports and correction requests to the appropriate developer(s) or documentation authors.
6. Hold meetings between the project manager and others to prioritize the issues and assign corrective work as necessary.
7. Developers and authors correct problems, perform appropriate unit and integration tests, and re-release software back to the Acceptance Test Process.
8. Repeat steps 1-3 as often as needed until all mandatory issues have been resolved and the system is approved and ready for release and production deployment.
9. The test team can sign off on a final test report, summarizing the final testing results and defect status, and sign a User Acceptance Sign-off form.
10. The project managers should get the project sponsor's signature on a Customer/User Acceptance Sign-off form.
11. The developer(s) and documentation authors then create the Production system, using the frozen development versions containing all the corrections implemented as a result of User Acceptance testing.

NOTE: In iterative development environments where incremental deliveries of a working system are to be made to users along the way, User Acceptance Testing should be employed on the appropriate scale for each release to users.

Template - User Acceptance Test Plan

Introduction

- **Purpose and Audience for the Document:** Note what application/system this UAT Plan applies to, and what teams and functional and operational groups this plan affects.
- **Acronyms Used:** Specify meaning of any acronyms or abbreviations used in the test plan.

Scope of the Testing

- **Types of Testing:** Address what types of testing will be included, for example:
 - **Operational Testing** ensures that requirements for data capture, data processing, data distribution and data archiving are met. This includes testing the system “at load” to ensure that load and performance aspects of transactions are handled.
 - **Functional Testing** ensures that all business functions are performed as per the business rules.
 - **Interface Testing** ensures that all business systems linked to the software system being tested pass and receive data or control as defined in the requirements specification.
 - **Usability Testing** ensures that the user is able to understand what to do on each screen, finds the operations flow smoothly, does not get confused, etc.
- **Related Materials to be Reviewed:** List accompanying documentation, etc., that should get ‘tested’ by users as well, e.g., Support Documentation Testing will be done to ensure that all accompanying materials needed by user to operate the system have also been exercised; for example, user manuals, installation manuals and online help.
- **Reference Documents:** List specification and design documents that the test team will test the functionality against, if applicable.

User Acceptance Testing Team Members and Responsibilities

- **Participants:** List who must be involved in the acceptance testing. Typical participants include:
 - Subject matter experts for the business processes or operations.
 - Business users involved in specifying requirements.

- QA manager.
- Key “typical” users and key “power” users as appropriate for the application. The intent is to ensure your test team involves a cross-section of users – those with different perspectives and abilities – so that acceptance testing truly yields acceptance by all user groups.

Also specify what backgrounds or preparation or knowledge the team members must have, for instance, understanding of certain business processes.

- **Testing Responsibilities:** Define who will be responsible for:
 - Developing specific test scenarios and cases.
 - Making assignments to specific test resources.
 - Executing the tests and recording results and defects, including judging completeness and accuracy of the business functionality – screens, reports, interfaces, and judging the user interface is acceptable for all groups/ types of users.
 - Recording defects or questions about the functionality tested.
- **UAT Team Training and Orientation:** Specify what training the test team members will need prior to testing. For example:
 - **Orientation to New Features:** Do users need training on the new system or new features in an existing system? To what degree do you intend to train them ahead of time vs. ‘test’ the usability of the system or certain features by letting them start testing with no or little training? (For some applications, it is a critical requirement that users be able to be ‘self-taught’ and learn and operate it without training beforehand.) Consider this question to determine how much training is appropriate prior to the start of your UAT.
 - **Orientation on Testing Process Itself:** For example, if the users are going to help write test cases, do they need brief training on how to do so? All team members might need an orientation session on how they will record test results and file defect reports.

Test Environment

- **Test Setup:** Specify the test environment – how many users, what systems, whether any special equipment is needed.
- **Databases:** Specify what test databases are needed to feed the application(s), how they should be populated and where they will reside.

Test Process

- **Setup of the Environment:** Specify who is responsible for setup and initialization of the test environment, as well as verifying that it is ready for UAT to start.
- **Delivery of Code into Acceptance Testing:** Specify how code will be released into the UAT environment and what testing and signoff must occur before the initial code enters UAT, and before any updates from defect corrections are admitted.
- **Test Execution and Records:** Specify what information will be recorded as tests are run, for example, conditions prior to starting a test case, actions taken during the test, test results, screen prints, updated checklists, copies of reports.
- **Defect Recording, Review:** Specify how each tester will record the failure of a particular test and create problem reports (defect reports). Refer to any related forms. Indicate how specification issues will be handled – e.g., through the change control process or as updates to the specification if they are simply clarifications. Specify who will meet and how often to prioritize issues.
- **Defect Severity Assignment:** Specify how issues will be classified to represent the relative severity, in terms of business/commercial impact, of a problem with the system found during testing. Example definitions:
 - **'Show Stopper':** It is impossible to continue with the testing because of the severity of this error/bug.
 - **Critical Problem:** Testing can continue, but we cannot go into production (live) with this problem.
 - **Major Problem:** Testing can continue, but this feature will cause severe disruption to business processes in live operation.
 - **Medium Problem:** Testing can continue and the system is likely to go live with only minimal departure from agreed business processes.
 - **Minor Problem:** Both testing and live operations may progress. This problem should be corrected, but little or no changes to business processes are envisioned.
 - **'Cosmetic' Problem:** e.g., colors, fonts, pitch size. Note, however, if such features are key to the business requirements, they will warrant a higher severity level.
- **Correction of Errors and Re-Testing:** Specify how any changed modules must be tested before admission to the UAT code base, including any regression testing.

Test Completion

- **Exit Criteria:** Users, the project sponsor and the project team must agree on the criteria for final acceptance of the system. For instance, agree upon the maximum number of errors (if any) allowed in the categories above at time of release.
- **Signoff Process:** Specify what formal signoffs are required and provide or reference related checklists or signature sheets.

Possible References or Attachments:

- Test setup and execution schedule
- Format for test cases
- Defect recording form to be used
- Signoff Form to be used
- Any related change control procedures

Administrative Information

Revision	Author	Date	Sections Affected	Change Summary
1.0		1/3/2009		

Current Version	1.0
Date	1/3/2009
Master Document Chapter Number	5
Document ID	230